

19



NL Octrooicentrum

11

2010546

12 A OCTROOIAANVRAAG

21 Aanvraagnummer: 2010546

51 Int.Cl.:
G06F 11/36 (2006.01)

22 Aanvraag ingediend: 29.03.2013

30 Voorrang:
30.03.2012 CN 201210102231

71 Aanvrager(s):
GE Medical Systems Global Technology
Company, LLC te Waukesha, Wisconsin,
Verenigde Staten van Amerika (US).

41 Aanvraag ingeschreven:
01.10.2013

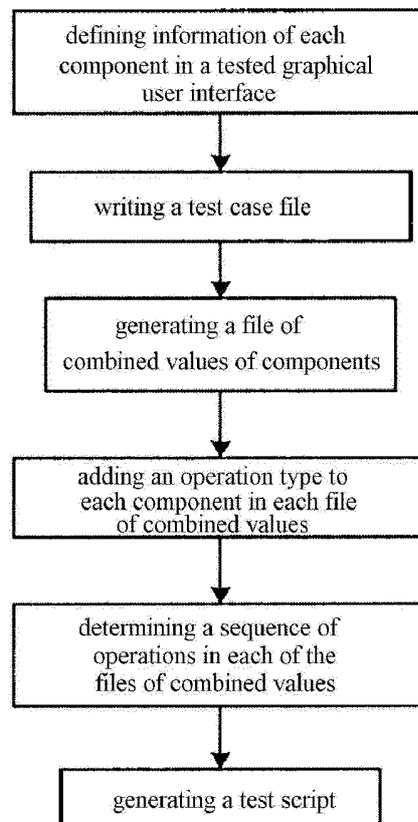
72 Uitvinder(s):
Yun Zhang te Beijing (CN).
Jing Luo te Beijing (CN).
Yuanji Wang te Beijing (CN).

43 Aanvraag gepubliceerd:
09.10.2013

74 Gemachtigde:
Ir. H.V. Mertens c.s. te Rijswijk.

54 Method and Apparatus for Automatically Generating A Test Script for a Graphical User Interface.

57 The present invention relates to the technical field of automatic test of a graphical user interface in the technical field of software test. The present invention provides a method for automatically generating a test script for a graphical user interface, which comprises: defining information of each component in a tested graphical user interface, writing a test case file, generating a file of combined values of components, adding an operation type to each component in each file of combined values, determining a sequence of operations in each of the files of combined values, and generating a test script, etc. Said method reduces the tester workload of manually programming test scripts and facilitates maintenance of test scripts.



NL A 2010546

Deze publicatie komt overeen met de oorspronkelijk ingediende stukken.

Method and Apparatus for Automatically Generating A Test Script for a Graphical User Interface

Technical Field

5

The present invention relates to the technical field of software test, in particular to the technical field of automatically generating a test script for a graphical user interface.

Background of the Invention

10

When using the prior art automation test tools to perform an automation test on a graphical user interface (GUI), testers need to manually program the corresponding test scripts according to the requirement of the test case. These automation test tools *per se* are only capable of automatically running these programmed test scripts. For example, the Windows-based automation test tool Winrunner and the Linux-based automation test tool Xrunner both belong to these types of automation test tools. Such automation test tools have the following problems:

1. It is difficult to program the test scripts manually. Different automation test tools have different programming styles and grammars, so testers have to be familiar with how to program qualified test scripts. Moreover, different test cases need to include different test scripts.

2. It is difficult to maintain the test scripts. Once the tested graphical user interface changes, all the test scripts will have to be manually modified.

Summary of the Invention

The technical problem to be solved by the present invention is to provide a method and apparatus for automatically generating a test script for a graphical user interface, which address the problem in the prior art that testers need to manually program test scripts and can hardly maintain the test scripts.

To solve this problem, a first aspect of the present invention lies in providing a method for automatically generating a test script for a graphical user interface, which comprises the following steps:

- 1) defining information of each component in a tested graphical user interface: defining an attribute of each component and a mutually constraint relation between said component and other components;
- 5 2) writing a test case file: storing values to be written into each component in the tested graphical user interface and test check point information in said file;
- 3) generating a file of combined values of components: combining values of all components in the test case file to obtain several files of combined values, each file
10 containing one possible combination;
- 4) adding an operation type to each component in each file of combined values: determining an operation type of each component value in the file of combined values according to the attribute of the component;
- 15 5) determining a sequence of operations in each of the files of combined values: rearranging the sequence of operations in the files of combined values according to the mutually constraint relation between the components; and
- 20 6) generating a test script: adding test check point information to the end of each file of combined values to obtain a test script file.

To solve the above-mentioned problem, a second aspect of the present invention, in combination with the first aspect, lies in providing a method for automatically
25 generating a test script for a graphical user interface, a range of value of an attribute of each component forms a set, which includes all possible operation types for said component.

To solve the above-mentioned problem, a third aspect of the present invention, in combination with the first aspect, lies in providing a method for automatically
30 generating a test script for a graphical user interface, a range of value of the mutually constraint relation of each component forms a set, which includes all input and output relations between said component and other components.

35 To solve the above-mentioned problem, a fourth aspect of the present invention, in combination with the first aspect, lies in providing a method for automatically generating a test script for a graphical user interface, the information of each

component further includes serial number, type and name of the component.

To solve the above-mentioned problem, the present invention also provides an apparatus for automatically generating a test script for a graphical user interface, which comprises: a module for defining information of each component in a tested graphical user interface, which defines an attribute of each component and a mutually constraint relation between said component and other components; a module for writing a test case file, which stores values to be written into each component in the tested graphical user interface and test check point information in said file; a module for generating a file of combined values of components, which combines values of all components in the test case file to obtain several files of combined values, each file containing one possible combination; a module for adding an operation type to each component value in each file of combined values, which determines an operation type of each component value in the file of combined values according to the attribute of the component; a module for determining a sequence of operations in each of the files of combined values, which rearranges the sequence of operations in the files of combined values according to the mutually constraint relation between the components; and a module for generating a test script, which adds test check point information to the end of each file of combined values to obtain a test script file.

20

Compared to the prior art, the method and apparatus for automatically generating a test script for a graphical user interface provided in the present invention achieve the following advantageous technical effects:

25 1. reducing the testers workload of manually programming test scripts; since test script files can be automatically generated according to the test case files, it is no longer necessary for testers to know the grammar for programming test scripts;

30 2. facilitating maintenance of test scripts; once the tested graphical user interface changes, new test script files can be regenerated automatically only by modifying the test case files.

Brief Description of the Drawings

35 Fig. 1 is a general flow chart of a method for automatically generating a test script for a graphical user interface provided by the present invention;

Fig. 2 is a schematic block diagram of an apparatus for automatically generating a test script for a graphical user interface provided by the present invention; and

Fig. 3 is a schematic drawing of a tested graphical user interface in a specific embodiment of the present invention.

Detailed Description of the Invention

The invention will be described further below with reference to the drawings and in conjunction with a specific embodiment.

In this embodiment, a graphical user interface shown in Fig. 3 is tested.

Step 1: defining information of each component in the tested graphical user interface in the following format:

$$W = \{id, type, name, attribute, constraint\}$$

wherein, id is the serial number of the component; type is the type of the component, e.g. text box, button, pull-down box, etc.; name is the name of the component; attribute is the attribute of the component, whose range of value is a set describing all possible operation types for the component, such as insertion, deletion, clicking, selection; constraint is the constraint relation between the component and other components, for example, a relation describing that the component is an input of another component.

It can be seen from Fig. 3 that said graphical user interface includes four components altogether, which are "Amount", "Rate", "Term" and "Calculate". For each component, the information thereof is defined according to the above-mentioned format as follows:

$$W1 = \{id1, textbox, Amount, insert/delete, w1 \rightarrow w4\}$$

$$W2 = \{id2, textbox, Rate, insert/delete, w2 \rightarrow w4\}$$

$$W3 = \{id3, pull-down\ box, Term, select, w3 \rightarrow w4\}$$

$$W4 = \{id4, button, Calculate, click, w4 \rightarrow Window2\}$$

wherein $w1 \rightarrow w4$, $w2 \rightarrow w4$, and $w3 \rightarrow w4$ mean that $w1$, $w2$ and $w3$ are inputs of $w4$ respectively; $w4 \rightarrow Window2$ means that $w4$ is outputted to another window, i.e. Window2, and contents displayed in Window2.

Step 2: writing a test case file, which includes the following information:

W1: 21, 0, -999

W2: 1.0, 6.48

W3: 1, 2, 3

W4: click, double click

5 Check Window2

In the test case file, values for each component form a line, which include values to be written into each of the components in the tested graphical user interface during the test, and the last line of the test case file specifies check point information of said test case.

10

Step 3: generating a file of combined values of the components. The component values in the test case file in step 2 are combined, and each possible combination is a file of combinations of component values. Examples of some of the combinations are as follows:

15

Combined Value File 1:

w1 21

w2 1.0

20 w3 1

w4 click

Combined Value File 2 :

w1 0

w2 1.0

25 w3 1

w4 click

Combined Value File 3 :

w1 -999

w2 1.0

30 w3 1

w4 click

Combined Value File 4 :

w1 -999

w2 6.48

35 w3 3

w4 double click

step 4: determining an operation type of each component value in each file of combined values. An operation type of each component value in the file of combined values is determined according to the attribute of the component defined in step 1, and then added to the file of combined values. The file of combined values having the operation type added thereto is as follows:

Combined Value File 1

insert w1 21
 10 insert w2 1.0
 select w3 1
 click w4 click

Combined Value File 2 :

insert w1 0
 15 insert w2 1.0
 select w3 1
 click w4 click

Combined Value File 3 :

insert w1 -999
 20 insert w2 1.0
 select w3 1
 click w4 click

Combined Value File 4 :

insert w1 -999
 25 insert w2 6.48
 select w3 3
 click w4 double click

step 5: determining a sequence of the operations in each of the files of combined values. According to the constraint relation between components defined in step 1, the outputs of w1, w2 and w3 are necessary inputs for w4, so corresponding operations for w1, w2 and w3 should be performed before operation of w4. Therefore, this rule should guide the re-ordering of the operations of the file of combined values obtained in step 4. In this embodiment, before the re-ordering, the sequence of operations in the file of combined values happens to meet the requirement of this rule, so the sequence

remains unchanged after the re-ordering.

Step 6: generating test scripts. The test check point information in the test case file written in step 2 is added to the end of each of the files of combined values obtained

5 in step 5 so as to obtain test script files as follows:

Test Script File 1:

insert w1 21

insert w2 1.0

select w3 1

10 click w4 click

check Window2

Test Script File 2 :

insert w1 0

insert w2 1.0

15 select w3 1

click w4 click

check Window2

Test Script File 3 :

insert w1 -999

20 insert w2 1.0

select w3 1

click w4 click

check Window2

Test Script File 4 :

25 insert w1 -999

insert w2 6.48

select w3 3

click w4 double click

check Window2

30

It can be seen from the above descriptions that the information defined in step 1 is provided by the developer and designer of the graphical user interface, and only the second step needs to be written by testers of the graphical user interface, which does not involve large workload. As for the next few steps that involve larger workload,

35 they can be performed automatically by a computer. Besides, when any modification

is made to the tested graphical user interface and a test has to be made again, only the test case file needs to be modified, and then the next four steps can be performed automatically by a computer so as to quickly generate the needed new test script.

CONCLUSIES

1. Werkwijze voor het automatisch genereren van een testinstructie voor een grafische gebruikersinterface, die de volgende stappen omvat:

5

1) het definiëren van informatie van elke component in een geteste grafische gebruikersinterface: het definiëren van een attribuut van elke component en een wederzijdse voorwaardelijke relatie tussen de component en andere componenten;

10

2) het schrijven van een testdossierbestand: het opslaan van waarden die moeten worden geschreven in elke component in de geteste grafische gebruikersinterface en testtafoetsinformatie in genoemd bestand;

15

3) het genereren van een bestand van gecombineerde waarden van componenten: het combineren van waarden van alle componenten in het testdossierbestand om verscheidene dossiers van gecombineerde waarden te verkrijgen, waarbij elk bestand is voorzien van één mogelijke combinatie;

20

4) het toevoegen van een bewerkingstype aan elk component in elk bestand van gecombineerde waarden: het bepalen van een bewerkingstype van elke componentwaarde in het bestand van gecombineerde waarden volgens het attribuut van de component;

25

5) het bepalen van een reeks bewerkingen in elk van de bestanden van gecombineerde waarden: het herschikken van de volgorde van bewerkingen in de bestanden van gecombineerde waarden volgens de wederzijds voorwaardelijke relatie tussen de componenten, en

30

6) het genereren van een testinstructie: het toevoegen van de testtafoetsinformatie aan het einde van elk bestand van gecombineerde waarden om een testinstructiebestand te bekomen.

2. Werkwijze volgens conclusie 1, waarbij een waardebereik van een attribuut van elke component een set vormt, die alle mogelijke bewerkingstypes voor genoemde component omvat.

35

3. Werkwijze volgens conclusie 1, waarbij een waardebereik van de wederzijdse voorwaardelijke relatie van ieder component een set vormt, die alle invoer- en uitvoerrelaties tussen genoemde component en andere componenten omvat.

4. Werkwijze volgens conclusie 1, waarbij de informatie van elke component verder serienummer, type en naam van de component omvat.

5. Inrichting voor het automatisch genereren van een testinstructie voor een grafische gebruikersinterface, omvattende: een module voor het definiëren van informatie van elke component in een geteste grafische gebruikersinterface, die een attribuut van elke component en een wederzijdse voorwaardelijke relatie tussen genoemde component en andere componenten definieert, een module voor het schrijven van een testdossierbestand, die waarden die moeten worden geschreven in elke component opslaat in de geteste grafische gebruikersinterface en testtafoetsinformatie in genoemd bestand, een module voor het genereren van een bestand van gecombineerde waarden van componenten, die waarden combineert alle componenten in het testdossierbestand om verscheidene dossiers van gecombineerde waarden te verkrijgen, waarbij elk bestand is voorzien van één mogelijke combinatie, een module voor het toevoegen van een bewerkingstype aan elk componentwaarde in elk bestand van gecombineerde waarden, die een bewerkingstype van elke componentwaarde bepaalt in het bestand van gecombineerd waarden volgens het attribuut van het onderdeel, een module voor de bepaling van een reeks bewerkingen in elk van de bestanden van gecombineerde waarden, die de volgorde van bewerkingen herschikt in de bestanden van gecombineerde waarden volgens de wederzijdse voorwaardelijke relatie tussen de componenten, en een module voor het genereren van een testinstructie, die testtafoetsinformatie toevoegt aan het einde van elk bestand van gecombineerde waarden om een testinstructiebestand te bekomen.

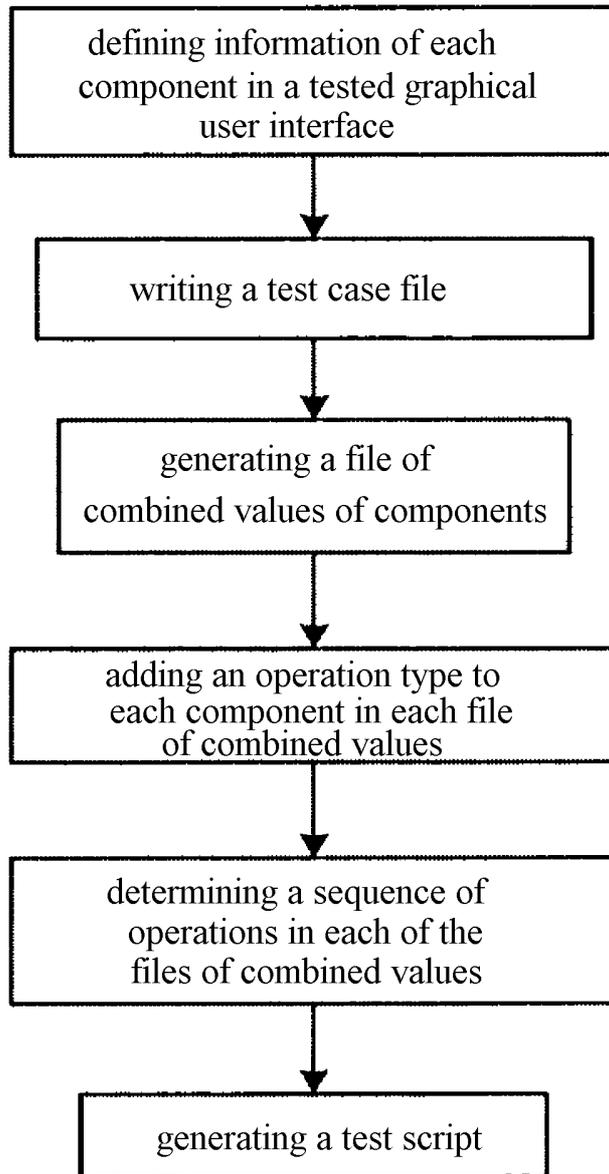


Fig. 1

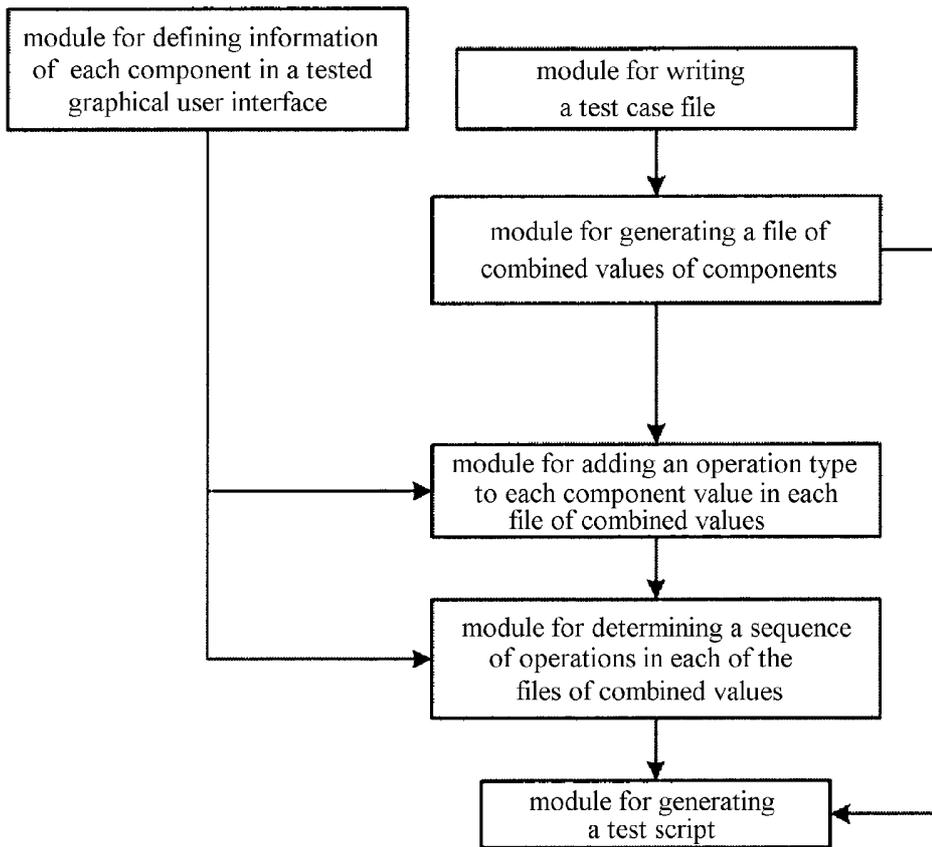


Fig. 2

Mortgage Calculator

Amount

Rate

Term ▼

Payement:

Fig. 3